

# Rajalakshmi Engineering College

Name: Loga Prabha GB  
Email: 240701290@rajalakshmi.edu.in  
Roll no: 240701290  
Phone: 9345046199  
Branch: REC  
Department: CSE - Section 3  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### REC\_Week 12\_Java\_Lamba Expressions\_MCQ

Attempt : 1  
Total Mark : 10  
Marks Obtained : 8

#### Section 1 : MCQ

1. Can a lambda expression in Java have a body with multiple statements?

**Answer**

Yes, if the statements are enclosed in curly braces

**Status : Correct**

**Marks : 1/1**

2. What is the syntax for a basic lambda expression in Java?

**Answer**

(parameters) -> expression

**Status : Correct**

**Marks : 1/1**

3. Which of the following is a valid lambda expression in Java?

**Answer**

(x) -> x \* 2

**Status : Wrong**

**Marks : 0/1**

4. What is a lambda expression in Java?

**Answer**

A way to define anonymous methods

**Status : Correct**

**Marks : 1/1**

5. Which functional interface in Java takes two arguments and returns a result?

**Answer**

BiFunction

**Status : Correct**

**Marks : 1/1**

6. Which functional interface is commonly used with lambda expressions in Java?

**Answer**

Runnable

**Status : Correct**

**Marks : 1/1**

7. Which of the following interfaces is NOT a functional interface in Java?

**Answer**

Comparator

**Status : Wrong**

**Marks : 0/1**

8. Can a lambda expression in Java have a body with multiple statements?

**Answer**

Yes, if the statements are enclosed in curly braces

**Status :** Correct

**Marks :** 1/1

9. What is the return type of a lambda expression in Java?

**Answer**

The return type is inferred from the context

**Status :** Correct

**Marks :** 1/1

10. Can a lambda expression have more than one parameter?

**Answer**

Yes, it can have multiple parameters

**Status :** Correct

**Marks :** 1/1

# Rajalakshmi Engineering College

Name: Loga Prabha GB  
Email: 240701290@rajalakshmi.edu.in  
Roll no: 240701290  
Phone: 9345046199  
Branch: REC  
Department: CSE - Section 3  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 12\_Q1

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Sabrina is working on a project that involves analyzing a set of numbers. In her exploration, she encounters scenarios where extracting even numbers and finding their sum is essential.

Create a program that calculates the sum of even numbers from a given array of integers using a lambda expression.

##### ***Input Format***

The first line of input consists of an integer N, representing the size of the array.

The second line consists of N space-separated integers, representing the elements of the array.

##### ***Output Format***

The output prints the sum of the even integers from the array.

Refer to the sample output for formatting specifications.

**Sample Test Case**

Input: 3

29 37 45

Output: 0

**Answer**

```
// You are using Java
import java.util.function.Predicate;
import java.util.Scanner;
class Main{
    public static void main(String[] args)
    {int a,n=0;
        Scanner in=new Scanner(System.in);
        Predicate<Integer> c=(b)->b%2==0;
        a=in.nextInt();
        for(int i=0;i<a;i++)
        {int b=in.nextInt();
            if(c.test(b))
            {n+=b;}
        }
        System.out.println(n);
    }
}
```

**Status : Correct**

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Loga Prabha GB  
Email: 240701290@rajalakshmi.edu.in  
Roll no: 240701290  
Phone: 9345046199  
Branch: REC  
Department: CSE - Section 3  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 12\_Q2

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Alex is learning about Java's functional interfaces and lambda expressions.

He wants to write a simple program that prints the square of each number in an array using a predefined functional interface.

Help Alex complete this task using the Consumer functional interface.

##### ***Input Format***

- The first line contains an integer N, the number of elements in the array.
- The second line contains N space-separated integers.

##### ***Output Format***

- Print the squares of all elements in the array, separated by a space.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: 4

1 2 3 4

Output: 1 4 9 16

### **Answer**

// You are using Java

```
import java.util.function.Consumer;
```

```
import java.util.*;
```

```
class Main{
```

```
    public static void main(String[] args)
```

```
    {int a;
```

```
    Scanner in=new Scanner(System.in);
```

```
    a=in.nextInt();
```

```
    int arr[]=new int[a];
```

```
    for(int i=0;i<a;i++)
```

```
    {arr[i]=in.nextInt();
```

```
    }
```

```
    Consumer<Integer> d=(l)->System.out.println(l*l);
```

```
    for(int s:arr)
```

```
    {d.accept(s);}}
```

```
}
```

**Status : Correct**

**Marks : 10/10**

# Rajalakshmi Engineering College

Name: Loga Prabha GB  
Email: 240701290@rajalakshmi.edu.in  
Roll no: 240701290  
Phone: 9345046199  
Branch: REC  
Department: CSE - Section 3  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 12\_Q3

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

In the mystical realm of programming, there exists a magical incantation to reveal hidden words.

Elara, the skilled enchantress, wishes to summon a word using her spell and then reverse its characters to uncover its enchanted reflection.

Write a program that uses the predefined functional interface `Supplier<String>` and a lambda expression to:

Supply (generate) a string, and

Display its reversed form.

**Input Format**



No input is required from the user.

The string must be supplied internally using a Supplier<String>.

### **Output Format**

Print the reversed version of the supplied string.

Refer to the sample output for formatting specifications.

### **Sample Test Case**

Input: Wizard!!

Output: !!draziW

### **Answer**

```
// You are using Java
import java.util.*;
import java.util.function.Supplier;
public class Main{
    public static void main(String[] args)
    {
        Scanner in=new Scanner(System.in);

        Supplier<String> s=()->in.next();
        StringBuilder str=new StringBuilder(s.get());
        System.out.println(str.reverse().toString());
    }
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Loga Prabha GB  
Email: 240701290@rajalakshmi.edu.in  
Roll no: 240701290  
Phone: 9345046199  
Branch: REC  
Department: CSE - Section 3  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### 2028\_REC\_OOPS using Java\_Week 12\_Q4

Attempt : 1  
Total Mark : 10  
Marks Obtained : 10

#### Section 1 : Coding

##### 1. Problem Statement

Abi is working on a text analysis project where she needs to categorize words based on their length.

Words that have three or fewer characters are considered "Short", while words with more than three characters are classified as "Long."

Write a Java program that takes a sentence as input, analyzes each word, and prints a list showing whether each word is "Short" or "Long."

Use the predefined functional interface `Function<String, String>` along with a lambda expression for categorization.

**Input Format**

A single line containing a sentence (words separated by spaces).

**Output Format**

- A single line with each word categorized as "Short" or "Long", separated by spaces.

Refer to the sample output for formatting specifications.

**Sample Test Case**

Input: I love my cat

Output: Short Long Short Short

**Answer**

```
// You are using Java
import java.util.*;
import java.util.function.Function;
class Main{
    public static void main(String[] args)
    {
        Scanner in=new Scanner (System.in);
        String[] k=in.nextLine().split(" ");
        Function<String,String> h=(g)->(g.length() <=3)?"Short":"Long";
        for(String r:k)
        {System.out.print(h.apply(r)+" ");}
    }
}
```

**Status : Correct**

**Marks : 10/10**